Seeded Database Matching Under Noisy Column Repetitions

Serhat Bakirtas, Elza Erkip

New York University

Information Theory Workshop 2022

Motivation

- Age of data collection.

Motivation

- Age of data collection.
- Potentially-sensitive data are made available for commercial and research purposes.

Motivation

- Age of data collection.
- Potentially-sensitive data are made available for commercial and research purposes.
  - User identities are removed: *Anonymization*.

Motivation

- Age of data collection.
- Potentially-sensitive data are made available for commercial and research purposes.
  - User identities are removed: *Anonymization*.
- Are anonymized data truly private?

Motivation

- Age of data collection.
- Potentially-sensitive data are made available for commercial and research purposes.
  - User identities are removed: *Anonymization*.
- Are anonymized data truly private?
- NO!

Motivation

- Age of data collection.
- Potentially-sensitive data are made available for commercial and research purposes.
  - User identities are removed: *Anonymization*.
- Are anonymized data truly private?
- NO!
  - Correlated public data $\rightarrow$ De-anonymization!

We Found Joe Biden's Secret Venmo. Here's Why That's A Privacy Nightmare For Everyone.

The peer-to-peer payments app leaves everyone from ordinary people to the most powerful person in the world exposed.


Ryan Mac
BuzzFeed News Reporter


Katie Notopoulos
BuzzFeed News Reporter


Ryan Brooks
BuzzFeed News Reporter

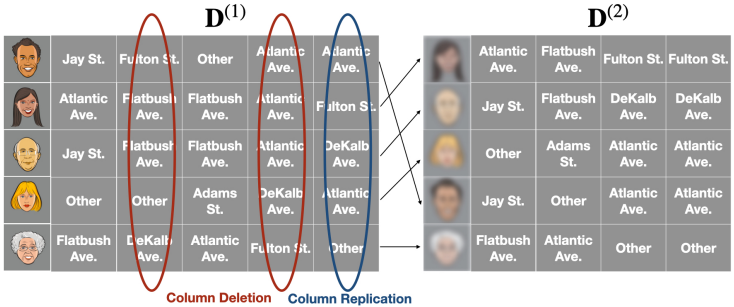

Logan McDonald
BuzzFeed Staff

Motivation: Our Work

- Anonymized databases containing *micro-information* shared and published routinely.
- Examples: Movie preferences, financial transactions data, location data, health records.

Motivation: Our Work

- Anonymized databases containing *micro-information* shared and published routinely.
- Examples: Movie preferences, financial transactions data, location data, health records.
- This work: Time-indexed data, *e.g.*, financial and location data
- Synchronization errors in time-indexed data: column repetitions

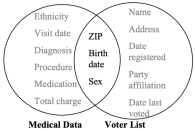


Practical Database Matching Attacks

- [Narayanan and Shmatikov, 2008]
  De-anonymization of Netflix Prize Database using IMDB data.

- [Sweeney, 2002]
  De-anonymization of medical databases using voter registration data.

- [Naini et al., 2012]
  User identification from geolocation data.



(a) Unlabeled histograms (Day 1)

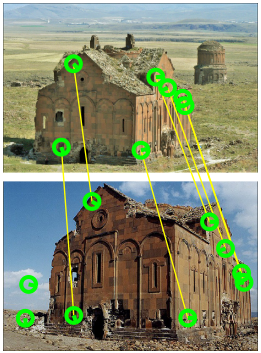
User	Location		
	Dorm.	Rest.	Lib.
?	75%	15%	10%
?	31%	30%	39%
?	15%	15%	70%
?	15%	65%	20%

(b) Labeled histograms (Day 2)

User	Location		
	Dorm.	Rest.	Lib.
John	33%	33%	34%
Jill	70%	20%	10%
Mary	15%	60%	25%
Mike	15%	20%	65%

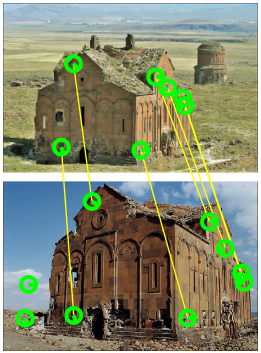
Database Matching: Other Applications

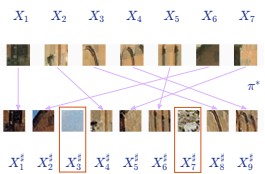
- Computer vision [Galstyan et al., 2021]



Database Matching: Other Applications

- Computer vision [Galstyan et al., 2021]

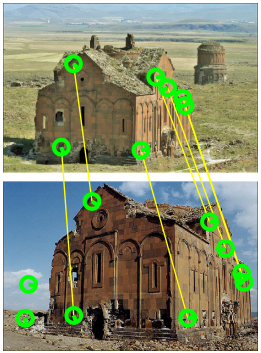




- Biological applications
  - DNA Sequencing [Blazewicz et al., 2002]

Database Matching: Other Applications

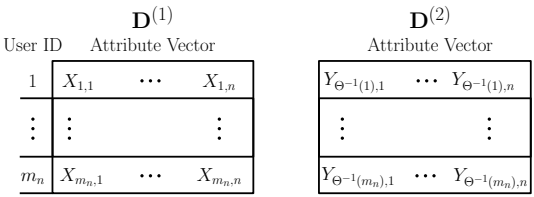
- Computer vision [Galstyan et al., 2021]



- Biological applications
  - DNA Sequencing [Blazewicz et al., 2002]
  - Single-cell data alignment [Chen et al., 2022]

Previous Works: Information-Theoretical Limits

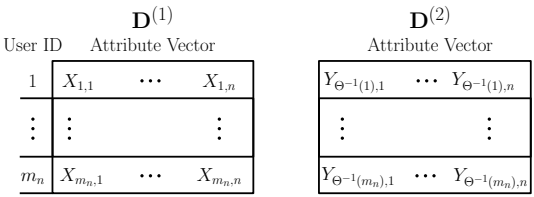
[Shirani, Garg, and Erkip, ISIT 2019]



- Databases as $m_n \times n$ random matrices: equal no. of labeled attributes (columns)
  - Matching rows $\sim f_{X^{(1),n},X^{(2),n}}$

Previous Works: Information-Theoretical Limits

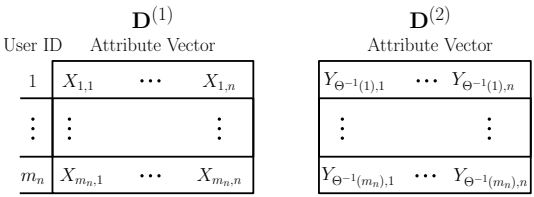
[Shirani, Garg, and Erkip, ISIT 2019]



- Databases as $m_n \times n$ random matrices: equal no. of labeled attributes (columns)
  - Matching rows $\sim f_{X^{(1),n},X^{(2),n}}$
- Database growth rate: $R = \lim_{n\to\infty} \frac{1}{n} \log m_n$

Previous Works: Information-Theoretical Limits

[Shirani, Garg, and Erkip, ISIT 2019]



- Databases as $m_n \times n$ random matrices: equal no. of labeled attributes (columns)
  - Matching rows $\sim f_{X^{(1),n},X^{(2),n}}$
- Database growth rate: $R = \lim_{n\to\infty} \frac{1}{n} \log m_n$
- Successful matching: $P_e \to 0$ as $n \to \infty$
- Database matching $\Leftrightarrow$ Channel decoding

Previous Works: Information-Theoretical Limits

- Objective: Given $(\mathbf{D}^{(1)}, \mathbf{D}^{(2)})$, find $\hat{\Theta}$ s.t.:

$$\Pr(\Theta(I) = \hat{\Theta}(I)) \to 1 \text{ as } n \to \infty,$$

  where $I \sim U(1, m_n)$.

- Almost all entries must be matched correctly.

Previous Works: Information-Theoretical Limits

- Objective: Given $(\mathbf{D}^{(1)}, \mathbf{D}^{(2)})$, find $\hat{\Theta}$ s.t.:

$$\Pr(\Theta(I) = \hat{\Theta}(I)) \to 1 \text{ as } n \to \infty,$$

  where $I \sim U(1, m_n)$.

- Almost all entries must be matched correctly.
  - In [Cullina et al., 2018], [Dai et al., 2019]: All entries must be matched correctly.

Previous Works: Information-Theoretical Limits

- Objective: Given $(\mathbf{D}^{(1)}, \mathbf{D}^{(2)})$, find $\hat{\Theta}$ s.t.:

$$\Pr(\Theta(I) = \hat{\Theta}(I)) \to 1 \text{ as } n \to \infty,$$

  where $I \sim U(1, m_n)$.

- Almost all entries must be matched correctly.
  - In [Cullina et al., 2018], [Dai et al., 2019]: All entries must be matched correctly.
- This allows us to
  - use information-theoretic tools,
  - work with arbitrary distributions.

Previous Works: Information-Theoretical Limits

[Bakirtas and Erkip, ISIT 2021]

- Database Matching Under Column Deletions.
  - Different numbers of attributes.
  - Attributes are unlabeled.

Previous Works: Information-Theoretical Limits

[Bakirtas and Erkip, ISIT 2021]

- Database Matching Under Column Deletions.
  - Different numbers of attributes.
  - Attributes are unlabeled.

- Sufficient conditions on database matching
  - Side information on the deletion locations.

Previous Works: Information-Theoretical Limits

[Bakirtas and Erkip, ISIT 2021]

- Database Matching Under Column Deletions.
  - Different numbers of attributes.
  - Attributes are unlabeled.
- Sufficient conditions on database matching
  - Side information on the deletion locations.
- Extracting this side information from a batch of correctly-matched rows (seeds).

Previous Works: Information-Theoretical Limits

[Bakirtas and Erkip, Asilomar 2022]

- Matching of Markov Databases Under Random Column Repetitions.
  - Different number of attributes (columns).
  - Attributes are unlabeled.
  - Markov rows.
    - Correlation among attributes
  - Noiseless setting.

Previous Works: Information-Theoretical Limits

[Bakirtas and Erkip, Asilomar 2022]

- Matching of Markov Databases Under Random Column Repetitions.
  - Different number of attributes (columns).
  - Attributes are unlabeled.
  - Markov rows.
    - Correlation among attributes
  - Noiseless setting.

- Repetition detection is possible without seeds in the noiseless setting.

Previous Works: Information-Theoretical Limits

[Bakirtas and Erkip, Asilomar 2022]

- Matching of Markov Databases Under Random Column Repetitions.
  - Different number of attributes (columns).
  - Attributes are unlabeled.
  - Markov rows.
    - Correlation among attributes
  - Noiseless setting.
- Repetition detection is possible without seeds in the noiseless setting.
- Complete characterization of the matching capacity in the noiseless setting.

This Talk: Seeded Database Matching Under Noisy Column Repetitions

1. The attributes are not labeled.
2. Databases do not have the same number of attributes.
   - Random column repetitions.

This Talk: Seeded Database Matching Under Noisy Column Repetitions

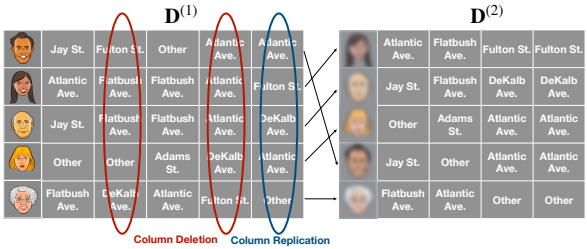
1. The attributes are not labeled.
2. Databases do not have the same number of attributes.
   - Random column repetitions.
3. The indices of the repeated columns are not known.

This Talk: Seeded Database Matching Under Noisy Column Repetitions

1. The attributes are not labeled.
2. Databases do not have the same number of attributes.
   - Random column repetitions.
3. The indices of the repeated columns are not known.
4. Repetition pattern is constant across the rows.

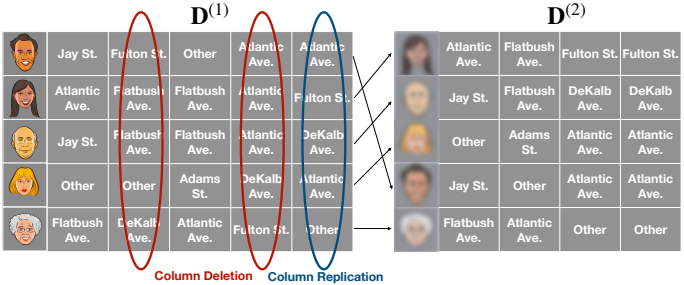
This Talk: Seeded Database Matching Under Noisy Column Repetitions

1. The attributes are not labeled.
2. Databases do not have the same number of attributes.
   - Random column repetitions.
3. The indices of the repeated columns are not known.
4. Repetition pattern is constant across the rows.
5. The retaining entries are noisy.



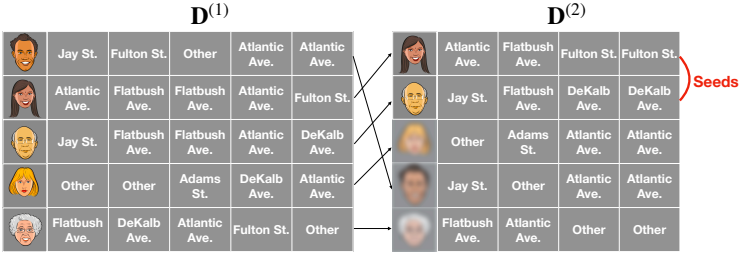
This Talk: Continued

- We have access to a batch of correctly-matched rows, *i.e.*, seeds.



This Talk: Continued

- We have access to a batch of correctly-matched rows, *i.e.*, seeds.



System Model

- $\mathbf{D}^{(1)}$: $m_n \times n$ random matrix with entries $X_{i,j} \overset{i.i.d.}{\sim} p_X$.

System Model

- $\mathbf{D}^{(1)}$: $m_n \times n$ random matrix with entries $X_{i,j} \overset{i.i.d.}{\sim} p_X$.
- Θ: uniform permutation of $[m_n]$.

System Model

- $\mathbf{D}^{(1)}$: $m_n \times n$ random matrix with entries $X_{i,j} \overset{i.i.d.}{\sim} p_X$.
- Θ: uniform permutation of $[m_n]$.
- Column repetition pattern: random vector $S^n = \{S_1, S_2, ..., S_n\}$ with $S_j \overset{i.i.d.}{\sim} p_S$.
  - $supp(p_S) = \{0, \ldots, s_{\max}\}$

System Model: Continued

System Model: Continued

$\mathbf{D}^{(2)}$: Obtained from $\mathbf{D}^{(1)}$ by

1. Row shuffling by Θ.

System Model: Continued

$\mathbf{D}^{(2)}$: Obtained from $\mathbf{D}^{(1)}$ by

1. Row shuffling by Θ.
2. Column deletion/replication by S^n.
   - Replicate the j^{th} column S_j times if $S_j > 0$.
   - Delete the j^{th} column if $S_j = 0$.

System Model: Continued

$\mathbf{D}^{(2)}$: Obtained from $\mathbf{D}^{(1)}$ by

1. Row shuffling by Θ.
2. Column deletion/replication by S^n.
   - Replicate the j^{th} column S_j times if $S_j > 0$.
   - Delete the j^{th} column if $S_j = 0$.
3. *i.i.d.* noise $p_{Y|X}$ on the retained entries.

System Model: Continued

$\mathbf{D}^{(2)}$: Obtained from $\mathbf{D}^{(1)}$ by

1. Row shuffling by Θ.
2. Column deletion/replication by S^n.
   - Replicate the j^{th} column S_j times if $S_j > 0$.
   - Delete the j^{th} column if $S_j = 0$.
3. *i.i.d.* noise $p_{Y|X}$ on the retained entries.
   - X and Y are not independent: $p_{Y|X} \neq p_Y$

System Model: Continued

- Seeds: Sub-databases $(\mathbf{G}^{(1)}, \mathbf{G}^{(2)})$ consisting of Λ_n pairs of correctly-matched rows.
  - $\Lambda_n = \Theta(n^d)$: Seed size
  - d: Seed order

System Model: Continued

- Seeds: Sub-databases $(\mathbf{G}^{(1)}, \mathbf{G}^{(2)})$ consisting of Λ_n pairs of correctly-matched rows.
  - $\Lambda_n = \Theta(n^d)$: Seed size
  - d: Seed order

- Achievable Database Growth Rate: Given $(\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, \mathbf{G}^{(1)}, \mathbf{G}^{(2)})$ with growth rate R and seed order d, $\exists \hat{\Theta}$ such that:

$$\Pr(\Theta(I) = \hat{\Theta}(I)) \to 1 \text{ as } n \to \infty,$$

  where $I \sim U(1, m_n)$.

- Matching Capacity:

$$C(d) \triangleq \sup\{R\text{: } R \text{ is achievable, given seed order } d.\}$$

System Model: Continued

- Seeds: Sub-databases $(\mathbf{G}^{(1)}, \mathbf{G}^{(2)})$ consisting of Λ_n pairs of correctly-matched rows.
  - $\Lambda_n = \Theta(n^d)$: Seed size
  - d: Seed order

- Achievable Database Growth Rate: Given $(\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, \mathbf{G}^{(1)}, \mathbf{G}^{(2)})$ with growth rate R and seed order d, $\exists \hat{\Theta}$ such that:

$$\Pr(\Theta(I) = \hat{\Theta}(I)) \to 1 \text{ as } n \to \infty,$$

  where $I \sim U(1, m_n)$.

- Matching Capacity:

$$C(d) \triangleq \sup\{R\text{: } R \text{ is achievable, given seed order } d.\}$$

- Goal: Given $p_X, p_{Y|X}, p_S, d$, characterize matching capacity $C(d)$.

This Talk: Objectives

1. What is the matching capacity?

This Talk: Objectives

1. What is the matching capacity?
2. Can we devise matching schemes which achieve this matching capacity?

This Talk: Objectives

1. What is the matching capacity?
2. Can we devise matching schemes which achieve this matching capacity?
3. Can we extract the repetition pattern from seeds?

This Talk: Objectives

1. What is the matching capacity?
2. Can we devise matching schemes which achieve this matching capacity?
3. Can we extract the repetition pattern from seeds?
4. If yes, how many seeds are sufficient?

Proposed Matching Scheme for Noisy Repetitions

- Exploit the identical repetition pattern across rows.

Proposed Matching Scheme for Noisy Repetitions

- Exploit the identical repetition pattern across rows.
  1. Find a permutation-invariant unique feature of the columns of $\mathbf{D}^{(2)}$.

Proposed Matching Scheme for Noisy Repetitions

- Exploit the identical repetition pattern across rows.
  1. Find a permutation-invariant unique feature of the columns of $\mathbf{D}^{(2)}$.
  2. By threshold testing these features, infer the noisy replicas.

Proposed Matching Scheme for Noisy Repetitions

- Exploit the identical repetition pattern across rows.
  1. Find a permutation-invariant unique feature of the columns of $\mathbf{D}^{(2)}$.
  2. By threshold testing these features, infer the noisy replicas.
  3. Using the seeds $(\mathbf{G}^{(1)}, \mathbf{G}^{(2)})$, extract the deletion pattern.

Proposed Matching Scheme for Noisy Repetitions

- Exploit the identical repetition pattern across rows.
  1. Find a permutation-invariant unique feature of the columns of $\mathbf{D}^{(2)}$.
  2. By threshold testing these features, infer the noisy replicas.
  3. Using the seeds $(\mathbf{G}^{(1)}, \mathbf{G}^{(2)})$, extract the deletion pattern.
  4. Group the noisy replica runs by introducing markers between the columns of $\mathbf{D}^{(2)}$.

Proposed Matching Scheme for Noisy Repetitions

- Exploit the identical repetition pattern across rows.
  1. Find a permutation-invariant unique feature of the columns of $\mathbf{D}^{(2)}$.
  2. By threshold testing these features, infer the noisy replicas.
  3. Using the seeds $(\mathbf{G}^{(1)}, \mathbf{G}^{(2)})$, extract the deletion pattern.
  4. Group the noisy replica runs by introducing markers between the columns of $\mathbf{D}^{(2)}$.
  5. Replace the deleted columns with erasure symbols in $\mathbf{D}^{(2)}$.

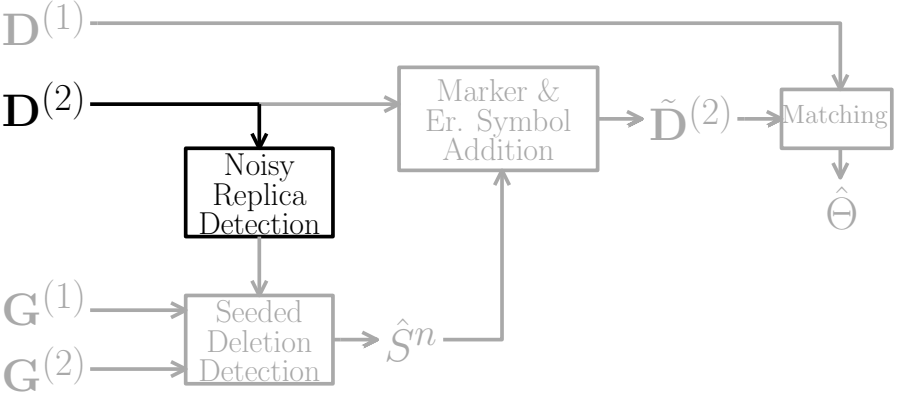
Proposed Matching Scheme for Noisy Repetitions

- Exploit the identical repetition pattern across rows.
  1. Find a permutation-invariant unique feature of the columns of $\mathbf{D}^{(2)}$.
  2. By threshold testing these features, infer the noisy replicas.
  3. Using the seeds $(\mathbf{G}^{(1)}, \mathbf{G}^{(2)})$, extract the deletion pattern.
  4. Group the noisy replica runs by introducing markers between the columns of $\mathbf{D}^{(2)}$.
  5. Replace the deleted columns with erasure symbols in $\mathbf{D}^{(2)}$.
  6. Perform a typicality-based rowwise matching.

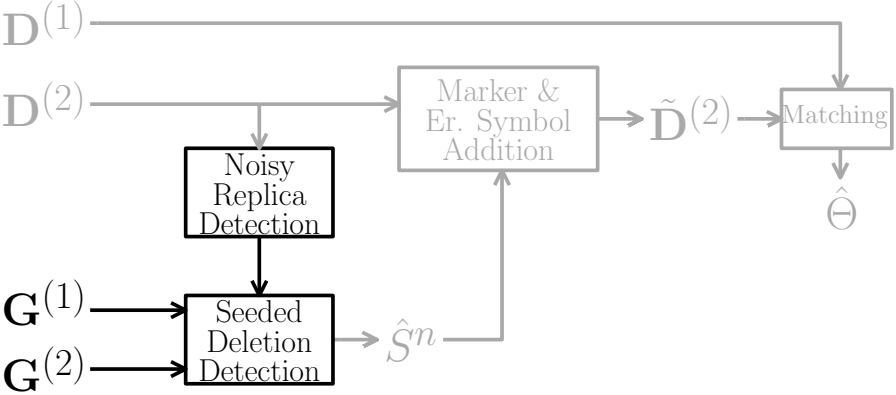
Proposed Matching Scheme for Noisy Repetitions

- Exploit the identical repetition pattern across rows.
  1. Find a permutation-invariant unique feature of the columns of $\mathbf{D}^{(2)}$.
  2. By threshold testing these features, infer the noisy replicas.
  3. Using the seeds $(\mathbf{G}^{(1)}, \mathbf{G}^{(2)})$, extract the deletion pattern.
  4. Group the noisy replica runs by introducing markers between the columns of $\mathbf{D}^{(2)}$.
  5. Replace the deleted columns with erasure symbols in $\mathbf{D}^{(2)}$.
  6. Perform a typicality-based rowwise matching.
- We will use the *Hamming distances between the consecutive columns of* $\mathbf{D}^{(2)}$ as the permutation-invariant feature.

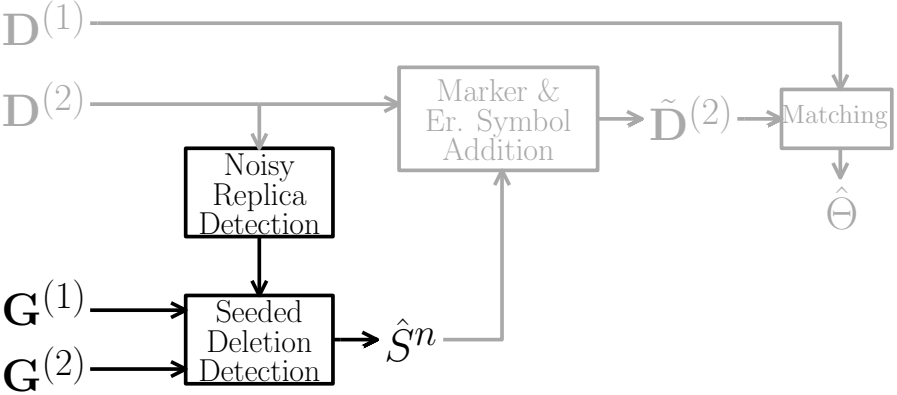
Proposed Matching Scheme



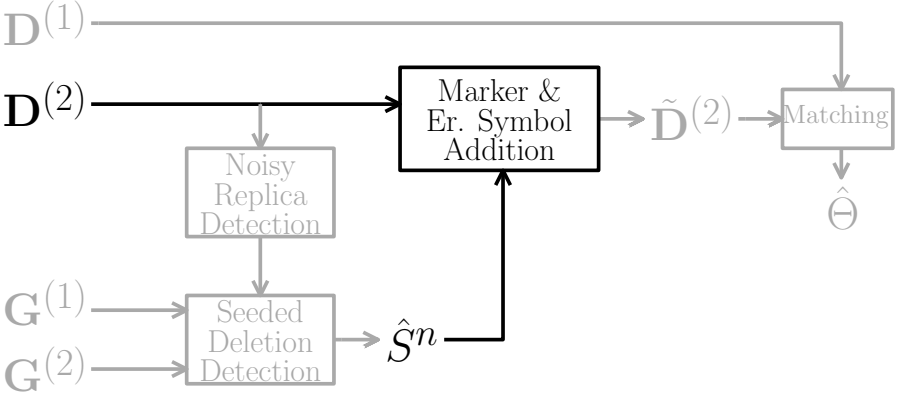
Proposed Matching Scheme



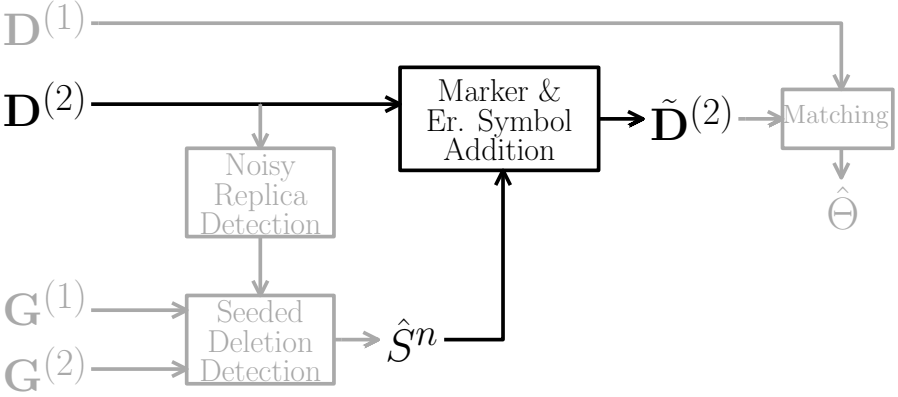
Proposed Matching Scheme



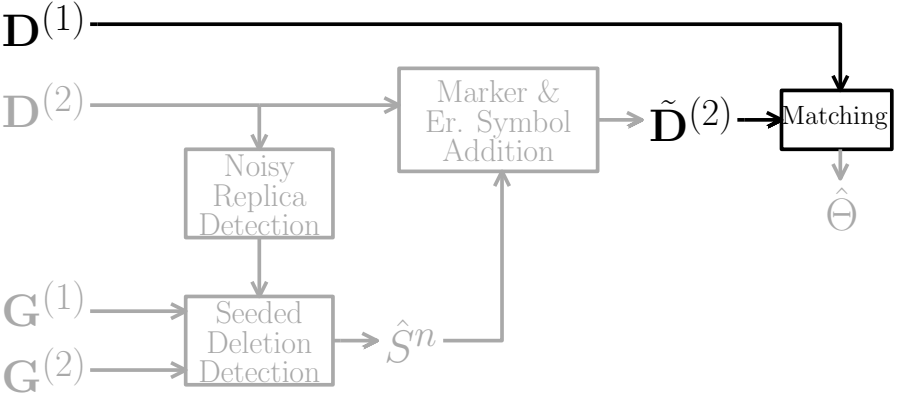
Proposed Matching Scheme



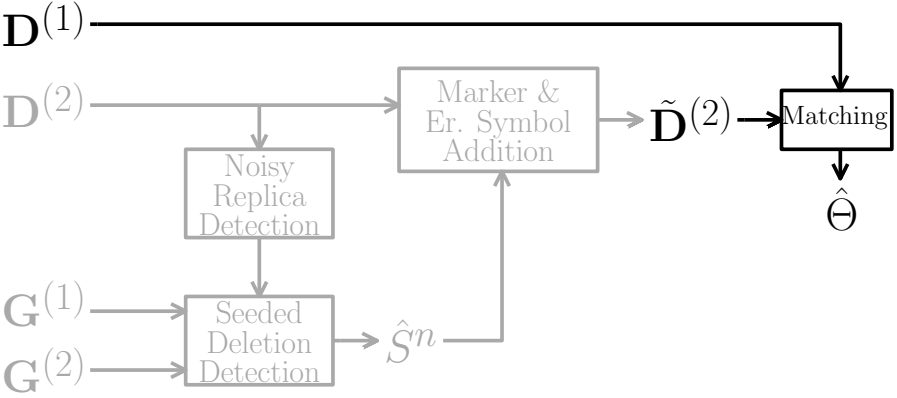
Proposed Matching Scheme



Proposed Matching Scheme



Proposed Matching Scheme



Noisy Replica Detection

- $C_j^{m_n}$: the j^{th} column of $\mathbf{D}^{(2)}$, $j = 1, \ldots, K$.

Noisy Replica Detection

- $C_j^{m_n}$: the j^{th} column of $\mathbf{D}^{(2)}$, $j = 1, \ldots, K$.

1. Choose an average threshold τ depending on $p_{X,Y}$.

Noisy Replica Detection

- $C_j^{m_n}$: the j^{th} column of $\mathbf{D}^{(2)}$, $j = 1, \ldots, K$.

1. Choose an average threshold τ depending on $p_{X,Y}$.
2. Compute the Hamming distances $d_H(C_j^{m_n}, C_{j+1}^{m_n})$ between $C_j^{m_n}$ and $C_{j+1}^{m_n}$, for $j \in [K-1]$.

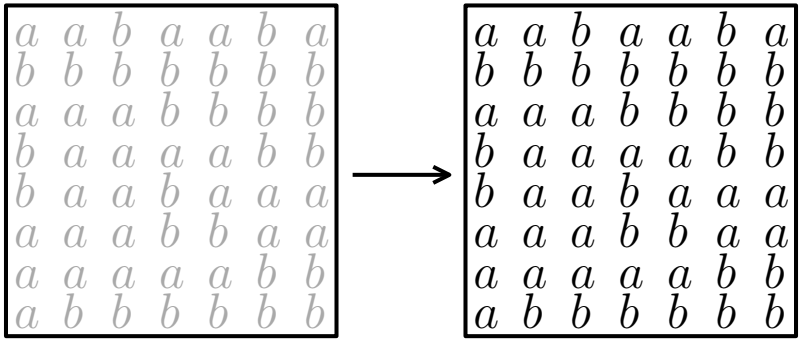
Noisy Replica Detection

- $C_j^{m_n}$: the j^{th} column of $\mathbf{D}^{(2)}$, $j = 1, \ldots, K$.

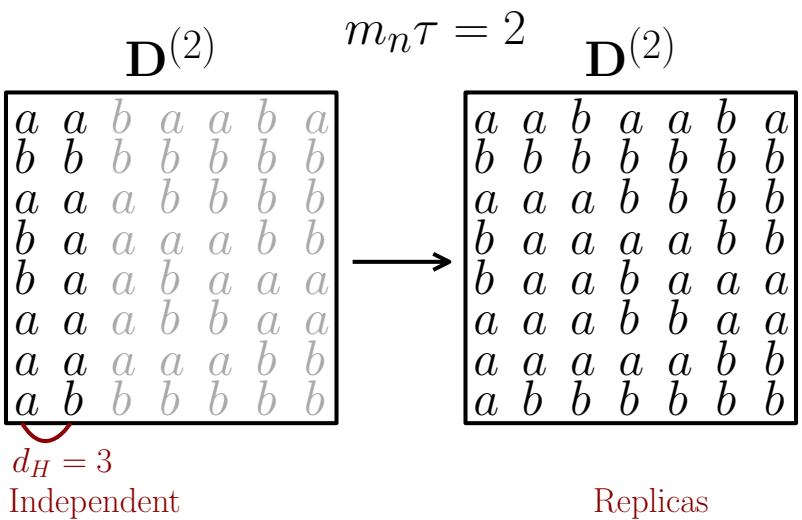
1. Choose an average threshold τ depending on $p_{X,Y}$.
2. Compute the Hamming distances $d_H(C_j^{m_n}, C_{j+1}^{m_n})$ between $C_j^{m_n}$ and $C_{j+1}^{m_n}$, for $j \in [K-1]$.
3. Declare $C_j^{m_n}$ and $C_{j+1}^{m_n}$ to be
   - noisy replicas, if $d_H(C_j^{m_n}, C_{j+1}^{m_n}) < m_n\tau$.
   - independent, if $d_H(C_j^{m_n}, C_{j+1}^{m_n}) \geq m_n\tau$.

Noisy Replica Detection: Example

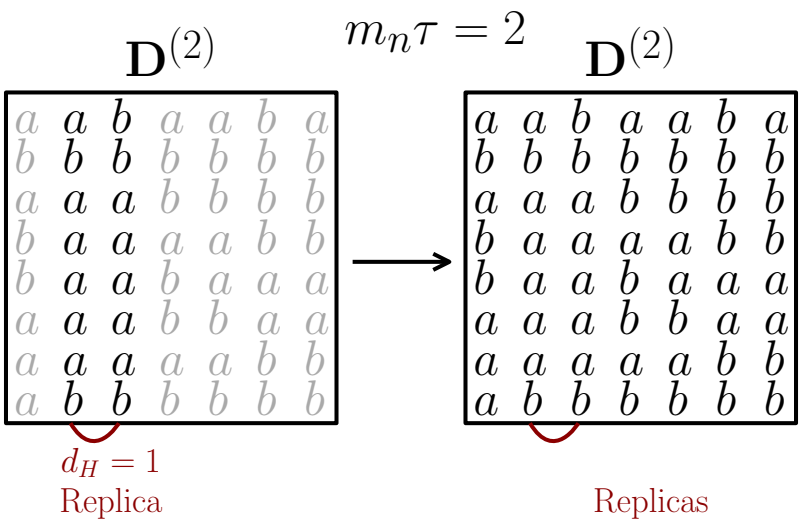
$\mathbf{D}^{(2)}$ $\quad m_n\tau = 2 \quad$ $\mathbf{D}^{(2)}$



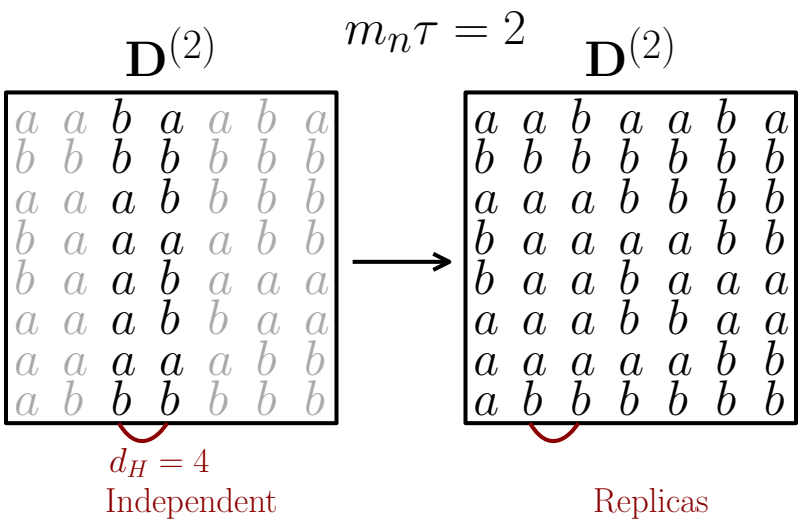
Noisy Replica Detection: Example



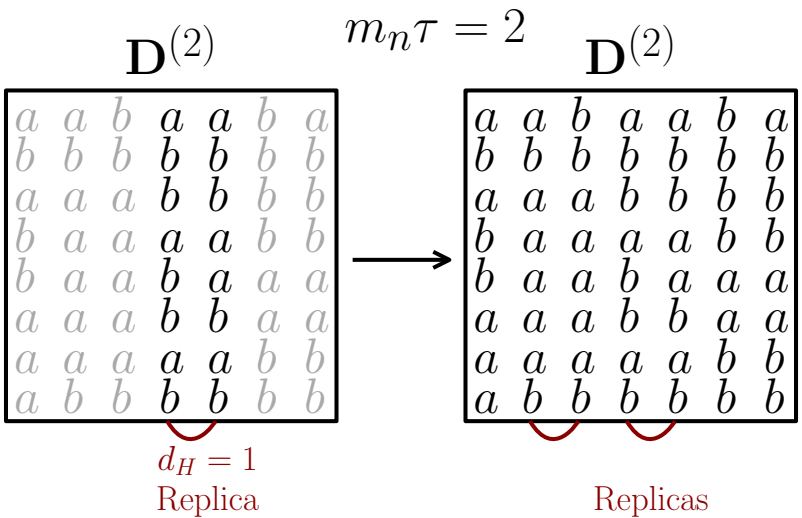
Noisy Replica Detection: Example



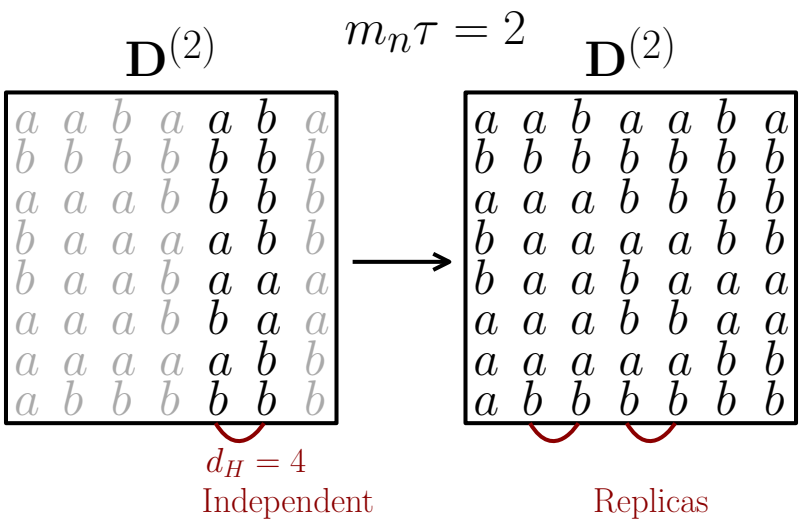
Noisy Replica Detection: Example



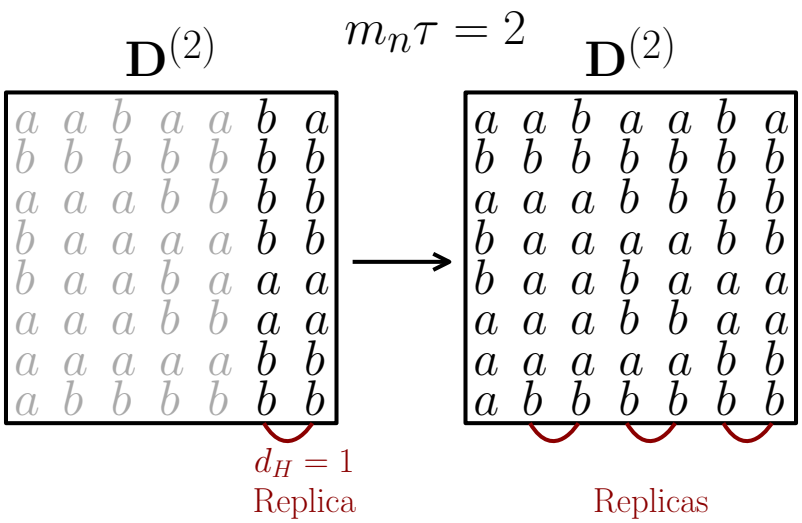
Noisy Replica Detection: Example



Noisy Replica Detection: Example



Noisy Replica Detection: Example



Noisy Replica Detection

Lemma

Let E_j denote the event that the aforementioned Hamming distance based algorithm fails to infer the correct relationship between the columns $C_j^{m_n}$ and $C_{j+1}^{m_n}$ of $\mathbf{D}^{(2)}$, $j = 1, \ldots, K-1$. Then

$$\Pr\Big(\bigcup_{j=1}^{K-1} E_j\Big) \to 0 \text{ as } n \to \infty$$

Noisy Replica Detection

Lemma

Let E_j denote the event that the aforementioned Hamming distance based algorithm fails to infer the correct relationship between the columns $C_j^{m_n}$ and $C_{j+1}^{m_n}$ of $\mathbf{D}^{(2)}$, $j = 1, \ldots, K-1$. Then

$$\Pr\Big(\bigcup_{j=1}^{K-1} E_j\Big) \to 0 \text{ as } n \to \infty$$

- No seeds required for noisy replica detection!

Noisy Replica Detection

Lemma

Let E_j denote the event that the aforementioned Hamming distance based algorithm fails to infer the correct relationship between the columns $C_j^{m_n}$ and $C_{j+1}^{m_n}$ of $\mathbf{D}^{(2)}$, $j = 1, \ldots, K-1$. Then

$$\Pr\Big(\bigcup_{j=1}^{K-1} E_j\Big) \to 0 \text{ as } n \to \infty$$

- No seeds required for noisy replica detection!
- m_n being exponential in n is enough.

Lemma: Sketch of Proof

- Let $(X_1, Y_1), (X_2, Y_2) \sim p_{X,Y}$.

Lemma: Sketch of Proof

- Let $(X_1, Y_1), (X_2, Y_2) \sim p_{X,Y}$.
- Define

$$p_0 \triangleq \Pr(Y_1 \neq Y_2 | X_1 \perp\!\!\!\perp X_2)$$
$$p_1 \triangleq \Pr(Y_1 \neq Y_2 | X_1 = X_2)$$

Lemma: Sketch of Proof

- Let $(X_1, Y_1), (X_2, Y_2) \sim p_{X,Y}$.
- Define

$$p_0 \triangleq \Pr(Y_1 \neq Y_2 | X_1 \perp\!\!\!\perp X_2)$$
$$p_1 \triangleq \Pr(Y_1 \neq Y_2 | X_1 = X_2)$$

- $d_H(C_j^{m_n}, C_{j+1}^{m_n})$ follows
  - $\text{Binom}(m_n, p_1)$ if $C_j^{m_n}$ and $C_{j+1}^{m_n}$ are noisy replicas.
  - $\text{Binom}(m_n, p_0)$ if $C_j^{m_n}$ and $C_{j+1}^{m_n}$ are independent.

Lemma: Sketch of Proof

- Let $(X_1, Y_1), (X_2, Y_2) \sim p_{X,Y}$.
- Define

$$p_0 \triangleq \Pr(Y_1 \neq Y_2 | X_1 \perp\!\!\!\perp X_2)$$
$$p_1 \triangleq \Pr(Y_1 \neq Y_2 | X_1 = X_2)$$

- $d_H(C_j^{m_n}, C_{j+1}^{m_n})$ follows
  - $\text{Binom}(m_n, p_1)$ if $C_j^{m_n}$ and $C_{j+1}^{m_n}$ are noisy replicas.
  - $\text{Binom}(m_n, p_0)$ if $C_j^{m_n}$ and $C_{j+1}^{m_n}$ are independent.
- Show $p_0 > p_1$ for any $p_{X,Y} \neq p_X p_Y$.

Lemma: Sketch of Proof

- Let $(X_1, Y_1), (X_2, Y_2) \sim p_{X,Y}$.
- Define

$$p_0 \triangleq \Pr(Y_1 \neq Y_2 | X_1 \perp\!\!\!\perp X_2)$$
$$p_1 \triangleq \Pr(Y_1 \neq Y_2 | X_1 = X_2)$$

- $d_H(C_j^{m_n}, C_{j+1}^{m_n})$ follows
  - $\text{Binom}(m_n, p_1)$ if $C_j^{m_n}$ and $C_{j+1}^{m_n}$ are noisy replicas.
  - $\text{Binom}(m_n, p_0)$ if $C_j^{m_n}$ and $C_{j+1}^{m_n}$ are independent.
- Show $p_0 > p_1$ for any $p_{X,Y} \neq p_X p_Y$.
- Choose any $\tau \in (p_1, p_0)$ bounded away from p_1 and p_0.

Lemma: Sketch of Proof

- Let $(X_1, Y_1), (X_2, Y_2) \sim p_{X,Y}$.
- Define

$$p_0 \triangleq \Pr(Y_1 \neq Y_2 | X_1 \perp\!\!\!\perp X_2)$$
$$p_1 \triangleq \Pr(Y_1 \neq Y_2 | X_1 = X_2)$$

- $d_H(C_j^{m_n}, C_{j+1}^{m_n})$ follows
  - $\text{Binom}(m_n, p_1)$ if $C_j^{m_n}$ and $C_{j+1}^{m_n}$ are noisy replicas.
  - $\text{Binom}(m_n, p_0)$ if $C_j^{m_n}$ and $C_{j+1}^{m_n}$ are independent.
- Show $p_0 > p_1$ for any $p_{X,Y} \neq p_X p_Y$.
- Choose any $\tau \in (p_1, p_0)$ bounded away from p_1 and p_0.
- Union bound on $\Pr(\bigcup_{j=1}^{K-1} E_j)$

Lemma: Sketch of Proof

- Let $(X_1, Y_1), (X_2, Y_2) \sim p_{X,Y}$.
- Define

$$p_0 \triangleq \Pr(Y_1 \neq Y_2 | X_1 \perp\!\!\!\perp X_2)$$
$$p_1 \triangleq \Pr(Y_1 \neq Y_2 | X_1 = X_2)$$

- $d_H(C_j^{m_n}, C_{j+1}^{m_n})$ follows
  - $\text{Binom}(m_n, p_1)$ if $C_j^{m_n}$ and $C_{j+1}^{m_n}$ are noisy replicas.
  - $\text{Binom}(m_n, p_0)$ if $C_j^{m_n}$ and $C_{j+1}^{m_n}$ are independent.
- Show $p_0 > p_1$ for any $p_{X,Y} \neq p_X p_Y$.
- Choose any $\tau \in (p_1, p_0)$ bounded away from p_1 and p_0.
- Union bound on $\Pr(\bigcup_{j=1}^{K-1} E_j)$
- Apply Chernoff bound to the summands

Lemma: Sketch of Proof

- Let $(X_1, Y_1), (X_2, Y_2) \sim p_{X,Y}$.
- Define

$$p_0 \triangleq \Pr(Y_1 \neq Y_2 | X_1 \perp\!\!\!\perp X_2)$$
$$p_1 \triangleq \Pr(Y_1 \neq Y_2 | X_1 = X_2)$$

- $d_H(C_j^{m_n}, C_{j+1}^{m_n})$ follows
  - $\text{Binom}(m_n, p_1)$ if $C_j^{m_n}$ and $C_{j+1}^{m_n}$ are noisy replicas.
  - $\text{Binom}(m_n, p_0)$ if $C_j^{m_n}$ and $C_{j+1}^{m_n}$ are independent.
- Show $p_0 > p_1$ for any $p_{X,Y} \neq p_X p_Y$.
- Choose any $\tau \in (p_1, p_0)$ bounded away from p_1 and p_0.
- Union bound on $\Pr(\bigcup_{j=1}^{K-1} E_j)$
- Apply Chernoff bound to the summands
- $\Theta(n)$ summands, each decaying exponentially with m_n.

Seeded Deletion Detection Algorithm

1. Perform noisy replica detection on $\mathbf{D}^{(2)}$.
2. Discard all-but-one of the replicas from $\mathbf{G}^{(2)}$ to obtain $\tilde{\mathbf{G}}^{(2)}$.
3. If necessary, apply a mapping Φ to the entries of $\tilde{\mathbf{G}}^{(2)}$ to obtain $\tilde{\mathbf{G}}^{(2)}_{\Phi}$
   - Φ satisfies

$$\Pr(\Phi(Y_1) \neq X_2) > \Pr(\Phi(Y_1) \neq X_1)$$

4. Perform an exhaustive search over all potential deletion patterns on $\mathbf{G}^{(1)}$.
5. For each deletion pattern I, compute the total Hamming distance $d_H(\tilde{\mathbf{G}}^{(1)}_I, \tilde{\mathbf{G}}^{(2)}_{\Phi})$ between $\tilde{\mathbf{G}}^{(1)}_I$ and $\tilde{\mathbf{G}}^{(2)}_{\Phi}$.
6. Output the deletion pattern $\hat{I}_{\text{del}}(\Phi)$, minimizing total Hamming distance between $\tilde{\mathbf{G}}^{(1)}_I$ and $\tilde{\mathbf{G}}^{(2)}_{\Phi}$

$$\hat{I}_{\text{del}}(\Phi) = \underset{I \subseteq [n]}{\arg\min}\, d_H(\tilde{\mathbf{G}}^{(1)}_I, \tilde{\mathbf{G}}^{(2)}_{\Phi})$$

Seeded Deletion Detection

Lemma

Let I_{del} be the underlying deletion pattern. Then there exists a bijective mapping Φ depending on $p_{X,Y}$ and for seed size $\Lambda_n \geq cnH_b(\delta)$,

$$\Pr\left(\hat{I}_{\text{del}}(\Phi) = I_{\text{del}}\right) \to 1 \text{ as } n \to \infty$$

where H_b denotes the binary entropy function, δ is the column deletion probability and c depends on $p_{X,Y}$.

Seeded Deletion Detection

Lemma

Let I_{del} be the underlying deletion pattern. Then there exists a bijective mapping Φ depending on $p_{X,Y}$ and for seed size $\Lambda_n \geq cnH_b(\delta)$,

$$\Pr\left(\hat{I}_{\text{del}}(\Phi) = I_{\text{del}}\right) \to 1 \text{ as } n \to \infty$$

where H_b denotes the binary entropy function, δ is the column deletion probability and c depends on $p_{X,Y}$.

- A seed size linear with the column size n is sufficient!

Seeded Deletion Detection

Lemma

Let I_{del} be the underlying deletion pattern. Then there exists a bijective mapping Φ depending on $p_{X,Y}$ and for seed size $\Lambda_n \geq cnH_b(\delta)$,

$$\Pr\left(\hat{I}_{\text{del}}(\Phi) = I_{\text{del}}\right) \to 1 \text{ as } n \to \infty$$

where H_b denotes the binary entropy function, δ is the column deletion probability and c depends on $p_{X,Y}$.

- A seed size linear with the column size n is sufficient!
- *i.e.*, a seed size logarithmic with the row size m_n is sufficient!

Lemma: Sketch of Proof

Detection:

- Union bound over all deletion patterns

$$\Pr\left(\hat{I}_{\text{del}}(\Phi) \neq I_{\text{del}}\right) \leq \sum_{I \subseteq [n], |I| = \hat{K}} \Pr(d_H(\tilde{\mathbf{G}}_I^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}) \leq d_H(\tilde{\mathbf{G}}_{I_{\text{del}}}^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}))$$

Lemma: Sketch of Proof

Detection:

- Union bound over all deletion patterns

$$\Pr\left(\hat{I}_{\text{del}}(\Phi) \neq I_{\text{del}}\right) \leq \sum_{I \subseteq [n], |I| = \hat{K}} \Pr(d_H(\tilde{\mathbf{G}}_I^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}) \leq d_H(\tilde{\mathbf{G}}_{I_{\text{del}}}^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}))$$

- Observe

$$d_H(\tilde{\mathbf{G}}_I^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}) - d_H(\tilde{\mathbf{G}}_{I_{\text{del}}}^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}) = M - N$$

$$M \sim \text{Binom}(\Lambda_n(\hat{K} - f(I, I_{\text{del}})), q_0(\Phi))$$

$$N \sim \text{Binom}(\Lambda_n(\hat{K} - f(I, I_{del})), q_1(\Phi))$$

Lemma: Sketch of Proof

Detection:

- Union bound over all deletion patterns

$$\Pr\left(\hat{I}_{\text{del}}(\Phi) \neq I_{\text{del}}\right) \leq \sum_{I \subseteq [n], |I| = \hat{K}} \Pr(d_H(\tilde{\mathbf{G}}_I^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}) \leq d_H(\tilde{\mathbf{G}}_{I_{\text{del}}}^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}))$$

- Observe

$$d_H(\tilde{\mathbf{G}}_I^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}) - d_H(\tilde{\mathbf{G}}_{I_{\text{del}}}^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}) = M - N$$
$$M \sim \text{Binom}(\Lambda_n(\hat{K} - f(I, I_{\text{del}})), q_0(\Phi))$$
$$N \sim \text{Binom}(\Lambda_n(\hat{K} - f(I, I_{del})), q_1(\Phi))$$

  - $f(I, I_{\text{del}})$: Overlap between the retention (non-deletion) patterns output by I and I_{del}.

Lemma: Sketch of Proof

Detection:

- Union bound over all deletion patterns

$$\Pr\left(\hat{I}_{\text{del}}(\Phi) \neq I_{\text{del}}\right) \leq \sum_{I \subseteq [n], |I| = \hat{K}} \Pr(d_H(\tilde{\mathbf{G}}_I^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}) \leq d_H(\tilde{\mathbf{G}}_{I_{\text{del}}}^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}))$$

- Observe

$$d_H(\tilde{\mathbf{G}}_I^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}) - d_H(\tilde{\mathbf{G}}_{I_{\text{del}}}^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}) = M - N$$
$$M \sim \text{Binom}(\Lambda_n(\hat{K} - f(I, I_{\text{del}})), q_0(\Phi))$$
$$N \sim \text{Binom}(\Lambda_n(\hat{K} - f(I, I_{del})), q_1(\Phi))$$

  - $f(I, I_{\text{del}})$: Overlap between the retention (non-deletion) patterns output by I and I_{del}.
- Apply Hoeffding's inequality to the summands.

Lemma: Sketch of Proof

Detection:

- Union bound over all deletion patterns

$$\Pr\left(\hat{I}_{\text{del}}(\Phi) \neq I_{\text{del}}\right) \leq \sum_{I \subseteq [n], |I| = \hat{K}} \Pr(d_H(\tilde{\mathbf{G}}_I^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}) \leq d_H(\tilde{\mathbf{G}}_{I_{\text{del}}}^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}))$$

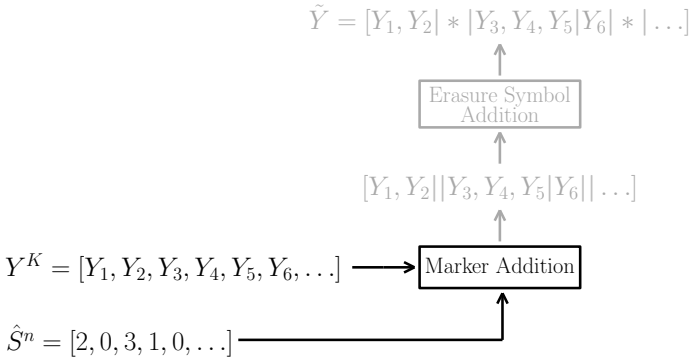
- Observe

$$d_H(\tilde{\mathbf{G}}_I^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}) - d_H(\tilde{\mathbf{G}}_{I_{\text{del}}}^{(1)}, \tilde{\mathbf{G}}_\Phi^{(2)}) = M - N$$
$$M \sim \text{Binom}(\Lambda_n(\hat{K} - f(I, I_{\text{del}})), q_0(\Phi))$$
$$N \sim \text{Binom}(\Lambda_n(\hat{K} - f(I, I_{del})), q_1(\Phi))$$

  - $f(I, I_{\text{del}})$: Overlap between the retention (non-deletion) patterns output by I and I_{del}.
- Apply Hoeffding's inequality to the summands.
- Sum over $f(I, I_{\text{del}})$ instead of I.

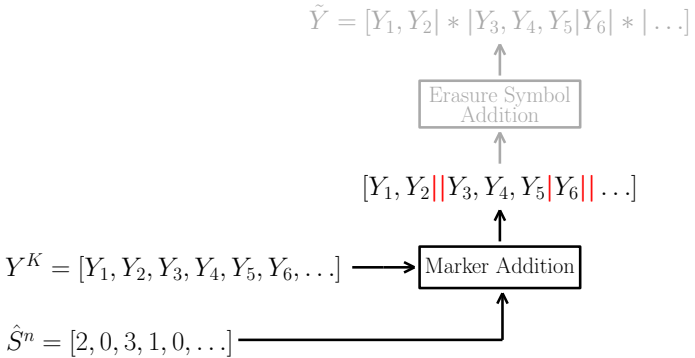
Marker & Erasure Symbol Addition

Y^K: a row of $\mathbf{D}^{(2)}$.



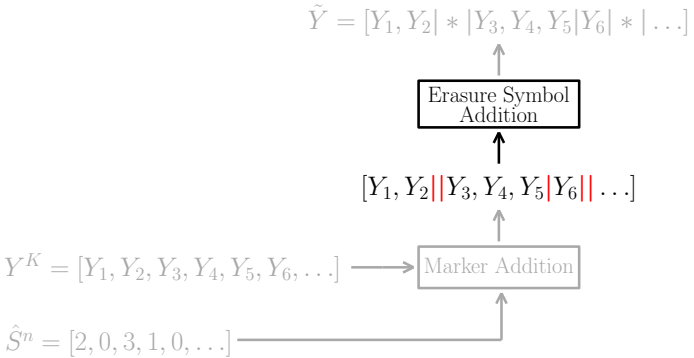
Marker & Erasure Symbol Addition

Y^K: a row of $\mathbf{D}^{(2)}$.



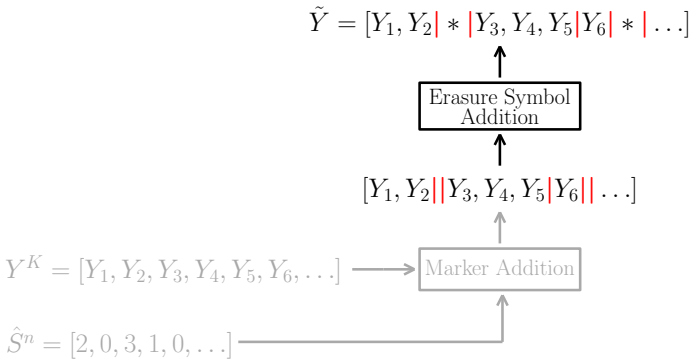
Marker & Erasure Symbol Addition

Y^K: a row of $\mathbf{D}^{(2)}$.



Marker & Erasure Symbol Addition

Y^K: a row of $\mathbf{D}^{(2)}$.



$\tilde{Y}$: the corresponding row of $\tilde{\mathbf{D}}^{(2)}$.

Main Result

Theorem: Main Result

Given a database distribution p_X, a column repetition distribution p_S and a noise distribution $p_{Y|X}$, for any seed order $d \geq 1$, the matching capacity is

$$C(d) = I(X; Y^S, S)$$

where $S \sim p_S$ and $Y^S = Y_1, \ldots, Y_S$ such that

$$\Pr(Y^S = y_1, \ldots, y_S | X = x) = \prod_{i=1}^{S} p_{Y|X}(y_i|x)$$

Main Result: Continued

- A logarithmic seed size is enough to infer S^n.

Main Result: Continued

- A logarithmic seed size is enough to infer S^n.
- Deleted columns do not offer any information.

Main Result: Continued

- A logarithmic seed size is enough to infer S^n.
- Deleted columns do not offer any information.
- Replicated columns offer additional information.

Main Result: Continued

- A logarithmic seed size is enough to infer S^n.
- Deleted columns do not offer any information.
- Replicated columns offer additional information.
  - Replication acts as a repetition code
  - With (randomly) varying length.

Main Result: Continued

- A logarithmic seed size is enough to infer S^n.
- Deleted columns do not offer any information.
- Replicated columns offer additional information.
  - Replication acts as a repetition code
  - With (randomly) varying length.
- We have a complete characterization of the matching capacity.

Conclusion

- Existence of an underlying repetition structure helps.

Conclusion

- Existence of an underlying repetition structure helps.
- Replicas can be inferred without any seeds.

Conclusion

- Existence of an underlying repetition structure helps.
- Replicas can be inferred without any seeds.
- A logarithmic seed size is sufficient for deletion detection.

Conclusion

- Existence of an underlying repetition structure helps.
- Replicas can be inferred without any seeds.
- A logarithmic seed size is sufficient for deletion detection.
- A tight bound on the achievable database growth rates.

Conclusion

- Existence of an underlying repetition structure helps.
- Replicas can be inferred without any seeds.
- A logarithmic seed size is sufficient for deletion detection.
- A tight bound on the achievable database growth rates.
- Converse result $\Rightarrow$ Insight into privacy-preserving publication of anonymized time-indexed microdata.

Conclusion

- Existence of an underlying repetition structure helps.
- Replicas can be inferred without any seeds.
- A logarithmic seed size is sufficient for deletion detection.
- A tight bound on the achievable database growth rates.
- Converse result $\Rightarrow$ Insight into privacy-preserving publication of anonymized time-indexed microdata.
- Ongoing Work:
  - More efficient deletion detection algorithms.

Conclusion

- Existence of an underlying repetition structure helps.
- Replicas can be inferred without any seeds.
- A logarithmic seed size is sufficient for deletion detection.
- A tight bound on the achievable database growth rates.
- Converse result $\Rightarrow$ Insight into privacy-preserving publication of anonymized time-indexed microdata.
- Ongoing Work:
  - More efficient deletion detection algorithms.
  - Extension to more general database distributions.

Conclusion

- Existence of an underlying repetition structure helps.
- Replicas can be inferred without any seeds.
- A logarithmic seed size is sufficient for deletion detection.
- A tight bound on the achievable database growth rates.
- Converse result $\Rightarrow$ Insight into privacy-preserving publication of anonymized time-indexed microdata.
- Ongoing Work:
  - More efficient deletion detection algorithms.
  - Extension to more general database distributions.
  - Database matching when the repetition pattern is not constant across rows.

Thank you! Q&A?

Seeded Database Matching Under Noisy Column Repetitions

Serhat Bakirtas, Elza Erkip

serhat.bakirtas@nyu.edu

Seeded Deletion Detection: A Problem

- This algorithm does not work for all $p_{X,Y}$!

Seeded Deletion Detection: A Problem

- This algorithm does not work for all $p_{X,Y}$!
- It depends on pairs of correlated entries in $\mathbf{G}^{(1)}$ and $\tilde{\mathbf{G}}^{(2)}$ having a higher probability of being equal than independent pairs.

Seeded Deletion Detection: A Problem

- This algorithm does not work for all $p_{X,Y}$!
- It depends on pairs of correlated entries in $\mathbf{G}^{(1)}$ and $\tilde{\mathbf{G}}^{(2)}$ having a higher probability of being equal than independent pairs.
- Formally, given $(X_1, Y_1) \sim p_{X,Y}$ and $(X_2, Y_1) \sim p_X p_Y$ it requires

$$\Pr(Y_1 = X_1) > \Pr(Y_1 = X_2)$$

Seeded Deletion Detection: A Problem

- This algorithm does not work for all $p_{X,Y}$!
- It depends on pairs of correlated entries in $\mathbf{G}^{(1)}$ and $\tilde{\mathbf{G}}^{(2)}$ having a higher probability of being equal than independent pairs.
- Formally, given $(X_1, Y_1) \sim p_{X,Y}$ and $(X_2, Y_1) \sim p_X p_Y$ it requires

$$\Pr(Y_1 = X_1) > \Pr(Y_1 = X_2)$$

- This is not true in general!

Seeded Deletion Detection: Example

$n = 6$, $\Lambda_n = 8$. $\mathfrak{X} = \{a, b\}$, $p_X(a) = p_X(b) = 0.5$, $p_{Y|X} \sim \text{BSC}(q)$, $q = 0.75$.

$$I_{del} = [1,\ 3]$$

$\mathbf{G}^{(1)}$

b	b	a	b	b	a
b	b	a	b	a	a
a	b	a	b	b	b
a	a	a	b	a	b
a	a	a	a	a	a
a	a	a	b	b	a
b	b	b	a	b	a
b	b	a	b	a	a

$\tilde{\mathbf{G}}^{(2)}$

b	b	b	a
b	b	a	a
b	b	b	b
a	b	a	b
a	a	a	a
a	b	b	a
b	a	b	a
b	b	a	a

$$\hat{I}_{del} = [2,\ 6]$$

Seeded Deletion Detection: Example

$n = 6$, $\Lambda_n = 8$. $\mathfrak{X} = \{a, b\}$, $p_X(a) = p_X(b) = 0.5$, $p_{Y|X} \sim \text{BSC}(q)$, $q = 0.75$.

$$I_{del} = [1,\ 3]$$

$\mathbf{G}^{(1)}$

b	b	a	b	b	a
b	b	a	b	a	a
a	b	a	b	b	b
a	a	a	b	a	b
a	a	a	a	a	a
a	a	a	b	b	a
b	b	b	a	b	a
b	b	a	b	a	a

$\tilde{\mathbf{G}}^{(2)}$

b	b	b	a
b	b	a	a
b	b	b	b
a	b	a	b
a	a	a	a
a	b	b	a
b	a	b	a
b	b	a	a

$$\hat{I}_{del} = [2,\ 6]$$

Seeded Deletion Detection: Example

$n = 6$, $\Lambda_n = 8$. $\mathfrak{X} = \{a, b\}$, $p_X(a) = p_X(b) = 0.5$, $p_{Y|X} \sim \text{BSC}(q)$, $q = 0.75$.

$$I_{del} = [1,\ 3]$$

$\mathbf{G}^{(1)}$

b	b	a	b	b	a
b	b	a	b	a	a
a	b	a	b	b	b
a	a	a	b	a	b
a	a	a	a	a	a
a	a	a	b	b	a
b	b	b	a	b	a
b	b	a	b	a	a

$\tilde{\mathbf{G}}^{(2)}$

a	b	a	b
a	a	b	b
a	a	a	a
b	a	b	a
b	a	b	b
a	a	a	b
a	b	a	b
a	a	b	b

$$\hat{I}_{del} = [2,\ 6]$$

Seeded Deletion Detection: Example

$n = 6$, $\Lambda_n = 8$. $\mathfrak{X} = \{a, b\}$, $p_X(a) = p_X(b) = 0.5$, $p_{Y|X} \sim \text{BSC}(q)$, $q = 0.75$.

$$I_{del} = [1,\ 3]$$

$\mathbf{G}^{(1)}$

b	b	a	b	b	a
b	b	a	b	a	a
a	b	a	b	b	b
a	a	a	b	a	b
a	a	a	a	a	a
a	a	a	b	b	a
b	b	b	a	b	a
b	b	a	b	a	a

$\tilde{\mathbf{G}}^{(2)}$

a	b	a	b
a	a	b	b
a	a	a	a
b	a	b	a
b	a	b	b
a	a	a	b
a	b	a	b
a	a	b	b

$$\hat{I}_{del} = [2,\ 6]$$

Seeded Deletion Detection: Example

$n = 6$, $\Lambda_n = 8$. $\mathfrak{X} = \{a, b\}$, $p_X(a) = p_X(b) = 0.5$, $p_{Y|X} \sim \text{BSC}(q)$, $q = 0.75$.

$$I_{del} = [1,\ 3]$$

$\mathbf{G}^{(1)}$

b	b	a	b	b	a
b	b	a	b	a	a
a	b	a	b	b	b
a	a	a	b	a	b
a	a	a	a	a	a
a	a	a	b	b	a
b	b	b	a	b	a
b	b	a	b	a	a

$\tilde{\mathbf{G}}^{(2)}$

a	b	a	b
a	a	b	b
a	a	a	a
b	a	b	a
b	a	b	b
a	a	a	b
a	b	a	b
a	a	b	b

$$\hat{I}_{del} = [2,\ 6]$$

Seeded Deletion Detection: Continued

- $\hat{I}_{\text{del}} \neq I_{\text{del}}$ in the above example.

Seeded Deletion Detection: Continued

- $\hat{I}_{\text{del}} \neq I_{\text{del}}$ in the above example.
- Observation: When $q > 0.5$, a symbol is more likely to flip, instead of staying the same.

Seeded Deletion Detection: Continued

- $\hat{I}_{\text{del}} \neq I_{\text{del}}$ in the above example.
- Observation: When $q > 0.5$, a symbol is more likely to flip, instead of staying the same.
- Solution: Flip the symbols, by applying the permutation $\Phi(a) = b$, $\Phi(b) = a$.

Seeded Deletion Detection: Continued

- $\hat{I}_{\text{del}} \neq I_{\text{del}}$ in the above example.
- Observation: When $q > 0.5$, a symbol is more likely to flip, instead of staying the same.
- Solution: Flip the symbols, by applying the permutation $\Phi(a) = b$, $\Phi(b) = a$.
- After applying Φ, we can use the aforementioned algorithm.

Seeded Deletion Detection: Example

$$I_{del} = [1,\ 3]$$

$\mathbf{G}^{(1)}$

b	b	a	b	b	a
b	b	a	b	a	a
a	b	a	b	b	b
a	a	a	b	a	b
a	a	a	a	a	a
a	a	a	b	b	a
b	b	b	a	b	a
b	b	a	b	a	a

$\tilde{\mathbf{G}}^{(2)}$

a	b	a	b
a	a	b	b
a	a	a	a
b	a	b	a
b	a	b	b
a	a	a	b
a	b	a	b
a	a	b	b

$$\hat{I}_{del}(\Phi) = [1,\ 3]$$

Seeded Deletion Detection: Example

$$I_{del} = [1,\ 3]$$

$\mathbf{G}^{(1)}$

b	b	a	b	b	a
b	b	a	b	a	a
a	b	a	b	b	b
a	a	a	b	a	b
a	a	a	a	a	a
a	a	a	b	b	a
b	b	b	a	b	a
b	b	a	b	a	a

$\tilde{\mathbf{G}}_{\Phi}^{(2)}$

b	a	b	a
b	b	a	a
b	b	b	b
a	b	a	b
a	b	a	a
b	b	b	a
b	a	b	a
b	b	a	a

$$\hat{I}_{del}(\Phi) = [1,\ 3]$$

Seeded Deletion Detection: Example

$$I_{del} = [1,\ 3]$$

$\mathbf{G}^{(1)}$

b	b	a	b	b	a
b	b	a	b	a	a
a	b	a	b	b	b
a	a	a	b	a	b
a	a	a	a	a	a
a	a	a	b	b	a
b	b	b	a	b	a
b	b	a	b	a	a

$\tilde{\mathbf{G}}_{\Phi}^{(2)}$

b	a	b	a
b	b	a	a
b	b	b	b
a	b	a	b
a	b	a	a
b	b	b	a
b	a	b	a
b	b	a	a

$$\hat{I}_{del}(\Phi) = [1,\ 3]$$

Seeded Deletion Detection: Example

$$I_{del} = [1,\ 3]$$

$\mathbf{G}^{(1)}$

b	b	a	b	b	a
b	b	a	b	a	a
a	b	a	b	b	b
a	a	a	b	a	b
a	a	a	a	a	a
a	a	a	b	b	a
b	b	b	a	b	a
b	b	a	b	a	a

$\tilde{\mathbf{G}}_{\Phi}^{(2)}$

b	a	b	a
b	b	a	a
b	b	b	b
a	b	a	b
a	b	a	a
b	b	b	a
b	a	b	a
b	b	a	a

$$\hat{I}_{del}(\Phi) = [1,\ 3]$$

Lemma: Sketch of Proof

Existence of Φ with desired property:

- Let

$$q_0(\Phi) \triangleq \Pr(\Phi(Y_1) \neq X_2)$$
$$q_1(\Phi) \triangleq \Pr(\Phi(Y_1) \neq X_1)$$

Lemma: Sketch of Proof

Existence of Φ with desired property:

- Let

$$q_0(\Phi) \triangleq \Pr(\Phi(Y_1) \neq X_2)$$
$$q_1(\Phi) \triangleq \Pr(\Phi(Y_1) \neq X_1)$$

- Explicitly write down the terms and show $\sum_{\Phi} q_0(\Phi) - q_1(\Phi) = 0$.

Lemma: Sketch of Proof

Existence of Φ with desired property:

- Let

$$q_0(\Phi) \triangleq \Pr(\Phi(Y_1) \neq X_2)$$
$$q_1(\Phi) \triangleq \Pr(\Phi(Y_1) \neq X_1)$$

- Explicitly write down the terms and show $\sum_{\Phi} q_0(\Phi) - q_1(\Phi) = 0$.
- Consider several one-cycle permutations over $\mathfrak{X}$ to show that

$$q_0(\Phi) - q_1(\Phi) = 0 \; \forall \Phi \Leftrightarrow p_{Y|X}(y|x) = p_Y(y) \; \forall (x, y) \in \mathfrak{X}^2$$

Lemma: Sketch of Proof

Existence of Φ with desired property:

- Let

$$q_0(\Phi) \triangleq \Pr(\Phi(Y_1) \neq X_2)$$
$$q_1(\Phi) \triangleq \Pr(\Phi(Y_1) \neq X_1)$$

- Explicitly write down the terms and show $\sum_{\Phi} q_0(\Phi) - q_1(\Phi) = 0$.
- Consider several one-cycle permutations over $\mathfrak{X}$ to show that

$$q_0(\Phi) - q_1(\Phi) = 0 \; \forall \Phi \Leftrightarrow p_{Y|X}(y|x) = p_Y(y) \; \forall (x, y) \in \mathfrak{X}^2$$

- Thus, as long as $p_{Y|X} \neq p_Y$,

$$\exists \Phi \; q_0(\Phi) > q_1(\Phi)$$